

**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) A method of generating a testbench for a representation of a device to be incorporated in a data processing apparatus, the testbench providing a test environment that represents one or more components of the data processing apparatus with which that device is to be coupled, the representation of the device being configurable based on configuration data specifying predetermined attributes of the one or more components, the method comprising the steps of:

- (a) receiving the configuration data used to configure the representation of the device;
- (b) generating the testbench with reference to the configuration data and a first set of templates defining the test environment; and
- (c) generating the representation of the device with reference to the configuration data and a second set of templates defining the representation of the device,

wherein the representation of the device is formed from constituent blocks and the second set of templates defines the representation of the device and its constituent blocks.

2. Canceled.

3. (previously presented) A method as claimed in Claim 1, further comprising the step of: providing a processing tool having access to the configuration data and the first and second sets of templates, said steps (b) and (c) being performed by the processing tool.

4. (original) A method as claimed in Claim 3, wherein the processing tool is operable independent of a language produced by the processing tool from each template.

5. (original) A method as claimed in Claim 1, wherein the representation of the device is provided in a first language type and at said step (b) a part of the testbench defined by a number of the templates in the first set is generated in a second language type different to the first language type.

6. (original) A method as claimed in Claim 5, wherein said first language type is a Register Transfer Language (RTL), and said second language type is a High level Verification Language (HVL).
7. (original) A method as claimed in Claim 1, wherein said device is a bus interconnect block.
8. (original) A method as claimed in Claim 7, further comprising the step of:  
employing a simulation tool to run a model of the data processing apparatus using the representation of the device and the testbench;  
wherein the first set of templates includes a master template defining a master engine coupled to a bus and operable during running of the model to generate test stimuli for input via the bus to the representation of the device.
9. (original) A method as claimed in Claim 8, wherein the master template includes a master monitor operable during running of the model to monitor signals on the bus to which the master engine is coupled.
10. (original) A method as claimed in Claim 9, wherein the testbench includes a scoreboard for checking data integrity within the model, and the master monitor is operable to output data to the scoreboard indicative of the signals on the bus to which the master engine is coupled.
11. (original) A method as claimed in Claim 8, wherein the master template includes a checker operable during running of the model to check that signals at an interface between the master engine and the bus to which the master engine is coupled conform to a protocol for that bus.
12. (original) A method as claimed in Claim 8, wherein the master engine is arranged to generate the test stimuli in a random manner.
13. (original) A method as claimed in Claim 7, further comprising the step of:

employing a simulation tool to run a model of the data processing apparatus using the representation of the device and the testbench;

wherein the first set of templates includes a slave template defining a slave engine coupled to a bus and operable during running of the model to generate response signals in reply to test stimuli received from the representation of the device.

14. (original) A method as claimed in Claim 13, wherein the slave template includes a slave monitor operable during running of the model to monitor signals on the bus to which the slave engine is coupled.

15. (original) A method as claimed in Claim 14, wherein the testbench includes a scoreboard for checking data integrity within the model, and the slave monitor is operable to output data to the scoreboard indicative of the signals on the bus to which the slave engine is coupled.

16. (original) A method as claimed in Claim 13, wherein the slave template includes a checker operable during running of the model to check that signals at an interface between the slave engine and the bus to which the slave engine is coupled conform to a protocol for that bus.

17. (original) A method as claimed in Claim 13, wherein the slave engine is arranged to generate the response signals in a random manner.

18. Canceled.

19. (original) A method as claimed in Claim 1, wherein there are a number of different component types, and the predetermined attributes specified by the configuration data indicate the component type for each of said one or more components.

20. (original) A method as claimed in Claim 19, wherein said device is a bus interconnect block, and wherein one of the component types is a master type, and for each of said one or more

components which is a master type, the predetermined attributes identify connections to any slave components within said one or more components that that master type component is connected to.

21. (original) A method as claimed in Claim 20, wherein the connections are identified as either connections to a local slave component not shared with other master components, or as connections through a bus matrix of the bus interconnect block to a shared slave component shared with one or more other master components.
22. (previously presented) A computer readable medium encoded with a computer program comprising code portions operable to control a computer to perform a method as claimed in Claim 1.
23. Canceled.
24. Canceled.
25. (currently amended) A system for generating a testbench for a representation of a device to be incorporated in a data processing apparatus, the testbench providing a test environment that represents one or more components of the data processing apparatus with which that device is to be coupled, the representation of the device being configurable based on configuration data specifying predetermined attributes of the one or more components, the system comprising:
  - logic operable to read the configuration data used to configure the representation of the device; and
  - generation logic operable to generate the testbench with reference to the configuration data and a first set of templates defining the test environment and to generate the representation of the device with reference to the configuration data and a second set of templates defining the representation of the device,  
wherein the representation of the device is formed from constituent blocks and the second set of templates defines the representation of the device and its constituent blocks.

26. Canceled.
27. (previously presented) A system as claimed in Claim 25, further comprising:  
a processing tool having access to the configuration data and the first and second sets of templates, the generation logic being provided by the processing tool.
28. (original) A system as claimed in Claim 27, wherein the processing tool is operable independent of a language produced by the processing tool from each template.
29. (original) A system as claimed in Claim 25, wherein the representation of the device is provided in a first language type, and during generation of the testbench by the generation logic a part of the testbench defined by a number of the templates in the first set is generated in a second language type different to the first language type.
30. (original) A system as claimed in Claim 29, wherein said first language type is a Register Transfer Language (RTL), and said second language type is a High level Verification Language (HVL).
31. (original) A system as claimed in Claim 25, wherein said device is a bus interconnect block.
32. (original) A system as claimed in Claim 31, further comprising:  
a simulation tool operable to run a model of the data processing apparatus using the representation of the device and the testbench;  
wherein the first set of templates includes a master template defining a master engine coupled to a bus and operable during running of the model to generate test stimuli for input via the bus to the representation of the device.
33. (original) A system as claimed in Claim 32, wherein the master template includes a master monitor operable during running of the model to monitor signals on the bus to which the master engine is coupled.

34. (original) A system as claimed in Claim 33, wherein the testbench includes a scoreboard for checking data integrity within the model, and the master monitor is operable to output data to the scoreboard indicative of the signals on the bus to which the master engine is coupled.
35. (original) A system as claimed in Claim 32, wherein the master template includes a checker operable during running of the model to check that signals at an interface between the master engine and the bus to which the master engine is coupled conform to a protocol for that bus.
36. (original) A system as claimed in Claim 32, wherein the master engine is arranged to generate the test stimuli in a random manner.
37. (original) A system as claimed in Claim 31, further comprising:  
a simulation tool operable to run a model of the data processing apparatus using the representation of the device and the testbench;  
wherein the first set of templates includes a slave template defining a slave engine coupled to a bus and operable during running of the model to generate response signals in reply to test stimuli received from the representation of the device.
38. (original) A system as claimed in Claim 37, wherein the slave template includes a slave monitor operable during running of the model to monitor signals on the bus to which the slave engine is coupled.
39. (original) A system as claimed in Claim 38, wherein the testbench includes a scoreboard for checking data integrity within the model, and the slave monitor is operable to output data to the scoreboard indicative of the signals on the bus to which the slave engine is coupled.
40. (original) A system as claimed in Claim 37, wherein the slave template includes a checker operable during running of the model to check that signals at an interface between the slave engine and the bus to which the slave engine is coupled conform to a protocol for that bus.

41. (original) A system as claimed in Claim 37, wherein the slave engine is arranged to generate the response signals in a random manner.

42. Canceled.

43. (original) A system as claimed in Claim 25, wherein there are a number of different component types, and the predetermined attributes specified by the configuration data indicate the component type for each of said one or more components.

44. (original) A system as claimed in Claim 43, wherein said device is a bus interconnect block, and wherein one of the component types is a master type, and for each of said one or more components which is a master type, the predetermined attributes identify connections to any slave components within said one or more components that that master type component is connected to.

45. (original) A system as claimed in Claim 44, wherein the connections are identified as either connections to a local slave component not shared with other master components, or as connections through a bus matrix of the bus interconnect block to a shared slave component shared with one or more other master components.

46. Canceled.

47. (new) A method of generating a testbench for a representation of a device to be incorporated in a data processing apparatus, the testbench providing a test environment that represents one or more components of the data processing apparatus with which that device is to be coupled, the representation of the device being configurable based on configuration data specifying predetermined attributes of the one or more components, the method comprising the steps of:

- (a) receiving the configuration data used to configure the representation of the device;
- (b) generating the testbench with reference to the configuration data and a first set of templates defining the test environment; and

(c) generating the representation of the device with reference to the configuration data and a second set of templates defining the representation of the device,  
wherein said device is a bus interconnect block,

the method further comprising employing a simulation tool to run a model of the data processing apparatus using the representation of the device and the testbench,

wherein the first set of templates includes a master template defining a master engine coupled to a bus and operable during running of the model to generate test stimuli for input via the bus to the representation of the device,

wherein the master template includes a master monitor operable during running of the model to monitor signals on the bus to which the master engine is coupled, and

wherein the testbench includes a scoreboard for checking data integrity within the model, and the master monitor is operable to output data to the scoreboard indicative of the signals on the bus to which the master engine is coupled.

48. (new) A method of generating a testbench for a representation of a device to be incorporated in a data processing apparatus, the testbench providing a test environment that represents one or more components of the data processing apparatus with which that device is to be coupled, the representation of the device being configurable based on configuration data specifying predetermined attributes of the one or more components, the method comprising the steps of:

- (a) receiving the configuration data used to configure the representation of the device;
- (b) generating the testbench with reference to the configuration data and a first set of templates defining the test environment; and
- (c) generating the representation of the device with reference to the configuration data and a second set of templates defining the representation of the device,

wherein said device is a bus interconnect block and the method further comprises the step of:

employing a simulation tool to run a model of the data processing apparatus using the representation of the device and the testbench,

wherein the first set of templates includes a slave template defining a slave engine coupled to a bus and operable during running of the model to generate response signals in reply to test stimuli received from the representation of the device,

wherein the slave template includes a slave monitor operable during running of the model to monitor signals on the bus to which the slave engine is coupled, and

wherein the testbench includes a scoreboard for checking data integrity within the model, and the slave monitor is operable to output data to the scoreboard indicative of the signals on the bus to which the slave engine is coupled.

49. (new) A method of generating a testbench for a representation of a device to be incorporated in a data processing apparatus, the testbench providing a test environment that represents one or more components of the data processing apparatus with which that device is to be coupled, the representation of the device being configurable based on configuration data specifying predetermined attributes of the one or more components, the method comprising the steps of:

- (a) receiving the configuration data used to configure the representation of the device;
- (b) generating the testbench with reference to the configuration data and a first set of templates defining the test environment; and
- (c) generating the representation of the device with reference to the configuration data and a second set of templates defining the representation of the device,

wherein there are a number of different component types, and the predetermined attributes specified by the configuration data indicate the component type for each of said one or more components,

wherein said device is a bus interconnect block,

wherein one of the component types is a master type, and for each of said one or more components which is a master type, the predetermined attributes identify connections to any slave components within said one or more components that that master type component is connected to, and

wherein the connections are identified as either connections to a local slave component not shared with other master components, or as connections through a bus matrix of the bus interconnect block to a shared slave component shared with one or more other master components.